

Scalable Adaptive Scan (SAS)

Anshuman Chandra, Rohit Kapur and Yasunari Kanzawa
Synopsys, Inc., 700 E. Middlefield Rd., Mountain View, CA

Abstract

Scan compression has emerged as the most successful solution to solve the problem of rising manufacturing test cost. Compression technology is not hierarchical in nature. Hierarchical implementations need test access mechanisms that keep the isolation between the different tests applied through the different compressors and decompressors. In this paper we discuss a test access mechanism for Adaptive Scan that addresses the problem of reducing test data and test application time in a hierarchical and low pin count environment. An active test access mechanism is used that becomes part of the compression schemes and unifies the test data for multiple CODEC implementations. Thus, allowing for hierarchical DFT implementations with flat ATPG.

1. Introduction

In the past decade the test industry has churned out a vast body of research in the area of scan compression [1]. Products are delivered by all the EDA companies delivering test data volume gains and test application time gains ranging from 1x to 100x. With roadmaps leading to 1000x this industry has spent most of its time on the quality of the results rather than on the flows to implement it. Today, as it stands, compression technology is seen as a flat implementation. However there is a significant need in the industry to have the DFT logic implemented within hierarchies even when the ATPG is going to be done on the entire design at one time (flat). To enable this, Test Access Mechanisms (TAM) are created to deliver and possibly schedule data to and from the multiple implementations of the decompressors and compressors [2-6]. The most common test access mechanism is to partition the scan-interface budget across each of the implementations within a design and providing direct connections to the compression logic from the external scan interface.

One of the compression schemes widely used in the industry is Adaptive Scan. This paper is about designing a test access mechanism for Adaptive Scan technology. The typical test access mechanism approach does not blend the highway to and from the hierarchy with the compression technology (they are passive). Our solution targets hierarchical DFT with flat ATPG which allows us to take the approach of blending the access mechanism into the compression technology itself (active test access

mechanism). The solution leverages data-pipelining methods known in the industry with flat compression technologies [7][8] to create a layer of logic over Adaptive Scan. We call this overall solution Scalable Adaptive Scan (SAS). SAS has the following unique characteristics:

1. Control data is pipelined and blended with scan-data to retain the powerful per-shift reconfigurability of Adaptive Scan.
2. A hierarchical solution is built such that the test access mechanism becomes part of the compression solution.
3. Bidirectional shifting is used to break dependencies created by extra sharing of signals in compression by the TAM layer.
4. Adaptive Scan's limitations on minimum pin usage are alleviated by the test access mechanism.

To begin with we describe the Adaptive Scan combinational solution and the existing concepts that are leveraged to create SAS. In the next section the technology with the above mentioned characteristics is described to create a compression solution for Adaptive Scan in a hierarchical environment. Finally the experimental results are presented with a focus on the most stringent configuration possible for a hierarchical implementation.

2. Adaptive Scan & Related Technologies

It is widely accepted that for higher number of internal chains to scan-interface ratios, sequential compressors and decompressors (CODECs) are far more efficient than combinational CODECs. However, wide adoption of the combinational compression technologies by the industry has proven otherwise. It has been shown that sequential CODECs are only slightly more efficient than the combinational ones [9]. This is especially true for the input side. The first methodology on using combinational decompressors with very few inputs was presented in [10]. This was followed by Illinois scan and its more efficient versions presented in [11][12].

Combinational compression schemes rely on the ability to supply and observe a large number of scan chains from a small interface. Figure 1 shows the impact of connecting three times the number of scan chains than the interface. A ratio of 3x more internal chains than the scan interface translates to 3x shorter chains and the corresponding reduction in the test data volume and test application time

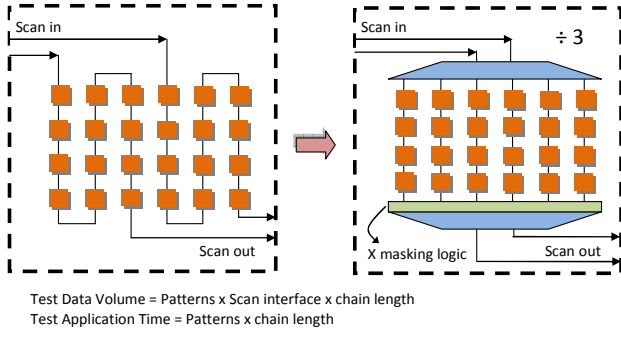


Figure 1: Decoupling of the scan-interface from the internal scan chains to allow for reduction in test data volume and test application time.

when compared with the scan implementation. Assuming no inefficiencies in the pattern generation process this is the fundamental mechanism behind the numerous research papers and the commercially available combinational scan compression technologies of today.

While combinational scan compression solutions are light weight and fit well in any design flow they do suffer from certain limitations:

1. Limited when the scan interface is too small.
2. Unable to generate more test data volume gains for high compression by generating tests from the decompressor itself.

Limited pins are not typically seen when implementing scan compression flat at the top level of the design. However, the industry seems to be moving to a flow where modules delivered from different parts of the same company are packaged with the compression solution. As a result of implementing many CODECs in a design a Test Access Mechanism is needed to deliver the connections to the top level. Because scan compression cannot be put together hierarchically a common solution is to partition the pin budget at the top level across all the CODECs and provide direct connections for each. Very quickly the CODECs are faced with a low pin problem. In this paper, we present active test access mechanism for many combinational CODECs in a hierarchical design. We focus our work on Adaptive Scan.

The Adaptive Scan compression technology presented first in [13] is a very widely used compression solution in the industry. Figure 2 shows the details of the CODEC that has been described in more detail in [14]. The input decompressor is constructed out of multiplexers that are supplied with scan data that ends up in the internal flip-flops of the design and control data that manages the multiplexer configurations. Therefore, each scan chain can be connected to a different scan input in any shift cycle using the control signals of the multiplexers. It should be noted that the architecture degenerates into a shared scan-

in architecture when the control data is held constant during the entire shift cycle.

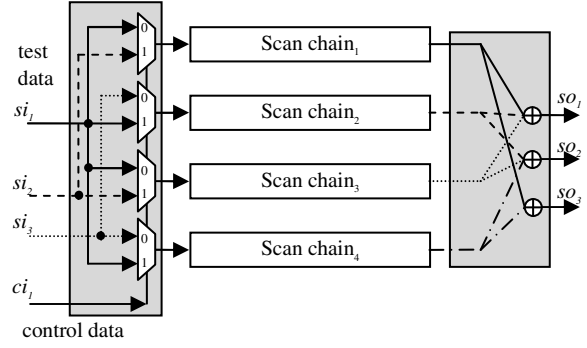


Figure 2: Adaptive scan decompressor with separate test data and control data inputs.

When the CODEC is built into a hierarchy, the additional delay added due to the long nets running from the scan ports to the CODEC often require additional pipelining flops to meet the chip timing. This is shown in Figure 3 (a) where a two stage pipelining has been implemented. *It is important to note that the additional pipeline flops do not affect the ATPG for the adaptive scan decompressor. Pipelining while sequential has been accepted as part of combinational solutions as it does not interfere with the way ATPG is performed in combinational compression. Pipelining maintains the simple relationships between signals of the combinational compression methods.* Dutta et. al. have shown [7] that these pipelining flops can be architected in a way to increase the data encoding efficiency of the combinational decompressor. In particular, all the scan inputs were considered as test data sources only and static connections were built from the pipeline flops to the decompressor to provide additional pseudo-scan-inputs (see Figure 3 (b)).

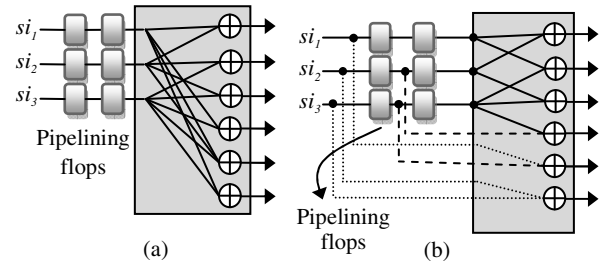


Figure 3: Combinational decompressor with two stage pipelining for input. (b) Use of pipelining to increase input bandwidth using pseudo inputs [7].

For output compression, Adaptive Scan technology uses a compressor built on the Stienen Triple system using exclusive-OR (XOR) gates [15] [14]. The compressor also contains masking logic to handle Xs in the response. The reader can find details on the compressor explained in the following paper [14]. Like the input, pipeline stages are added to the output for timing issues. Related research in convolutional compactors has used additional flops with

the XOR gates to provide a space and time tradeoff of the compacted test responses [8].

When considering industrial design-for-test flows, a good hierarchical solution is as important as the compression solution itself. Figure 4 shows two such ways in which the Adaptive Scan technology is inserted in the designs in hierarchical flows. As is evident from the diagram, the hierarchical flows require all the ports at the core level to be brought up to the top level. Therefore if the user is constrained by the number of pins at the top level, he/she has to either compromise on the compression QoR at the core level by targeting lower compression with fewer pins or multiplex the top level scan pins between two compressed cores and test one core at a time. However, a hierarchical solution is desired that gives good compression with fewer scan pins and enables testing of all cores in parallel.

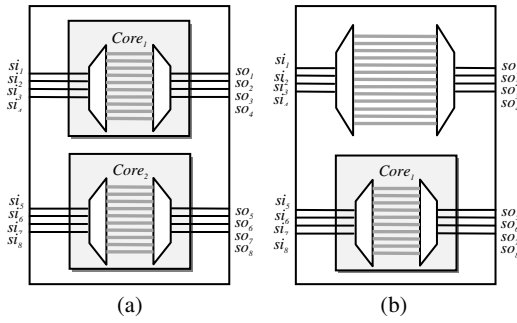


Figure 4: Hierarchical flows for combinational compression: (a) Two compressed cores integrated at the top level. (b) One compressed core with a top level CODEC.

3. Hierarchical Scalable Solution

While hierarchical flows have multiple implementations of Adaptive Scan in the design, we focus our initial description to a single CODEC of Adaptive Scan. Since the test access mechanism being used interferes with the compression technology it is important to understand the implications. The test access mechanism itself is simple and is easily extendable to multiple scan-in/out and multiple hierarchies.

Since pipeline flip-flops are already considered part of a combinational solution we create a test access mechanism that is built out of a shift register (pipeline of flip-flops). Since prior work has shown how pipelining flip-flops [7] can be used to increase bandwidth, we make the pipe-line for the test access mechanism interfere with the decompressor. Similarly on the output side we blend the pipelining flip-flops with the compressor of Adaptive Scan to get similar effects to the convolutional compaction scheme that was shown for a single compressor. On an individual Adaptive Scan implementation the most constrained solution for inputs and outputs is shown in Figure 5.

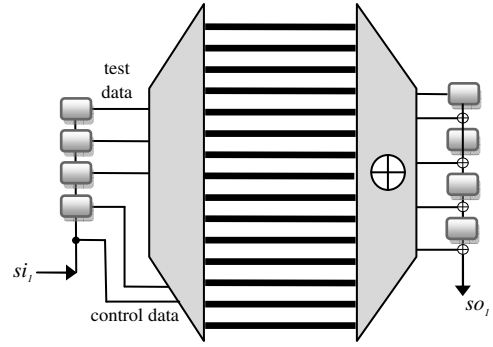


Figure 5: Scalable adaptive scan architecture with a shift register at the input and the output.

3.1 Scan-in operation

Adaptive Scan has two types of inputs, the *control inputs* and the *data inputs*. The shift data is required to first encounter the control inputs of Adaptive Scan then the data inputs. Furthermore if all the control configurations are not defined for Adaptive Scan some combinational logic would be needed between the test access mechanism and the control inputs to ensure that the illegal configuration cannot occur.

Feeding the control data through a shift register allows the Adaptive Scan architecture to retain the advantages of being able to change the modes dynamically on per shift basis, albeit with slightly less degree of freedom. For example, if the decompressor has four modes, then with the shift register solution, the modes can only change in a gray code fashion rather than having a full freedom of changing from one mode to any of the four modes. Since the values appearing on the control inputs are values that occur during shift of the test access mechanism all configurations of the multiplexers need to be defined.

The test access mechanism is driven off the same clock and scan enable signals as the compression structures. Hence when a value is being shifted in the test access mechanism the same value also moves into the compression logic. Figure 6 attempts to show this effect of the test access mechanism on Adaptive Scan. Adaptive Scan is built out of multiplexers in the decompressor that degenerates to a shared-scan-in architecture when the control inputs are held constant. The dependencies in the values that appear in the scan-cells caused by the sharing of the single scan-in are shown in Figure 6(a) for one decompressor mode. Flip-flops with the same color denote flip-flops that always have the same values if there is no inversion in the scan chains. Figure 6 (b) shows the effect of a shift register on dependencies in the scan-chains when the scan signals are driven by a shift register. We observe that the vertical data dependency translates into diagonal dependencies for a given direction of data flow in the shift register. The use of the shift register increases the number of shift operations it takes for a value to go from the scan-input to the most distant flip-flop. In this example the shift

time increases from 5 cycles (chain length) to 8 cycles because of the 3 bit test access mechanism.

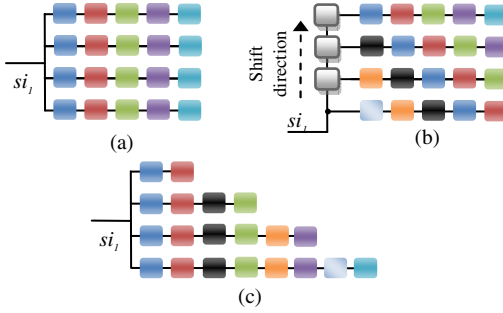


Figure 6: (a) Data dependencies in simple one-to-four shared fanout configuration. (b) Data dependencies for one-to-four shared fanout configuration with a shift register. (c) Modeling a shift register scan-in into a one-to-four shared scan-in fanout configuration.

Figure 6 (c) is a representation of the dependencies seen in Figure 6 (b) when implemented as a shared scan-in equivalent. If the scan chain length is l , and test access mechanism shift register length is l^{SR} , the scan flop at $j = 1$ and $(l + l^{SR})$ have no data dependency with any other scan cell, where j is the shift cycle. Similarly, flops at $j = 2$ and $(l + l^{SR} - 1)$ have two cell dependency and so on till the diagonal dependency is equal to the number of scan chains. Therefore the shift register based architecture shown in Figure 6 (b) can be easily transformed into a shared-scan-in architecture without a shift register as shown in Figure 6 (c). The data dependencies in Figure 6 (c) are exactly the same as in Figure 6 (b) and the scan shift cycle is 8 in both cases. While the figure represents one mode of Adaptive Scan with no scan-in sharing, other modes of adaptive scan can also be transformed into static single input shared scan-in models and scan-in sharing could also be modeled. Figure 7 (a) shows another equivalent of the interactions of the test access mechanism with the decompressor when implemented with pipelining flip-flops and a single scan-in.

3.2 Scan-out operation

To drive the response to a single output, the test access mechanism is a shift register at the output as well. However, unlike input side, we need to include an XOR between the shift register elements so as not to lose the response bit from the previous element (see Figure 5). The adaptive scan compressor is designed out of purely combinational gates that are added to the end of the internal scan chain outputs. The compressor is based on Steiner Triple Systems and is designed to tolerate up to two Xs per shift with no loss of observability on any other chain while enabling direct diagnosis and minimizing error aliasing. Due to page restrictions, the reader is referred to [14] where the construction of the combinational compressor has been discussed in detail. The important point to note here is that due to the design of the

combinational compressor and the connections made to the shift register on the output, the combined compressor scheme becomes a convolutional compactor. As discussed in [8], the convolutional compactors have exactly the same structural design and have been shown to be very efficient for compacting test responses.

The transformation shown in Figure 6 for the input side also holds true for the output side when the test access mechanism is connected. In this case the colors would represent the values being XORed with each other.

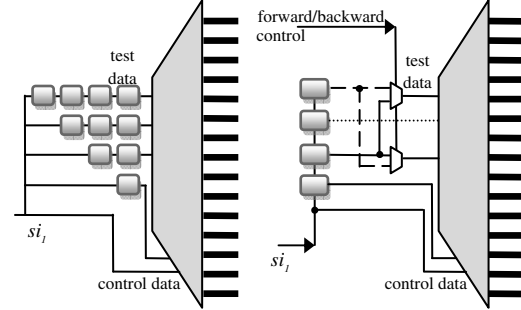


Figure 7: (a) Implementing shift register equivalent at the input using pipelining. (b) Simple multiplexor based control to reverse the data flow direction in the shift register.

3.3 Reducing Dependencies Further

As higher compression is targeted, the scan cell dependencies increase non-linearly. Therefore, any simple enhancement that can help reduce/change the cell dependencies is of great benefit to improve the compression QoR. One such simple methodology is to have a static signal which can reverse the direction of data flow into the compressor and decompressor (Figure 7 (b) shows the decompressor). As shown in Figure 6 (b), the diagonal dependencies created are dependent on the direction of data flow. By reversing that in the data signals portion of the decompressor, the scan cell dependencies are completely changed and as shown in the results section, this can greatly improve the compression QoR. On the output side, reversing of the data flow causes the X canceling effect to completely change from the forward direction data flow, thereby enhancing response observation. Depending on the area overhead and pins available, more such static shift register configurations can be made, though with diminishing returns.

3.4 Hierarchical Implementation

It is very common in the industry to reuse old designs or insert intellectual property cores (IPs) in the current generation of designs. If these cores have compression already built into them, today there are not many solutions that can actually help scale the compression for such cores and satisfy the high compression values required with low scan pin count. For example if a core was designed with a 10X compression CODEC, it is not straight forward to extract 50X compression for the same core with smaller number of pins at the top level. TAM design and compression has been studied previously and the common

approach has been to keep the two as independent as possible [16][17][18][19]. The shift register based TAM solution provides a very simple way to actually reduce scan pin count at the top level and scale the compression at the same time. As shown in Figure 8, the number of scan pins required at the top level reduces from 30 pins (15 scanin + 15 scanout) to just 4 pins. This is significant specially, as shown in Section 4, if much higher compression can also be achieved. This also raises questions on what is the best way to design the CODEC at the core level and the TAM design at the top level. A good solution may require relaxing the compression ratio at the core level, using bigger shift registers at the top level, or connecting CODEC inputs of a core to either a dedicated shift register with an independent input or sharing it with another CORE (see Figure 8).

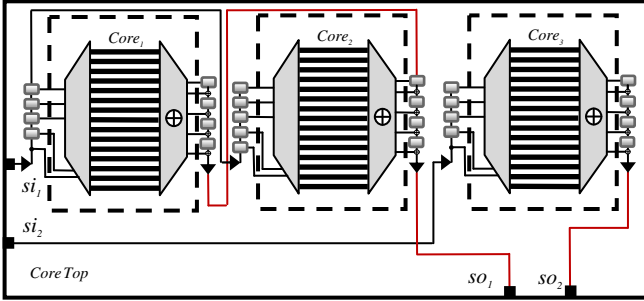


Figure 8: Hierarchical implementation of active TAM to achieve high compression with low scan pin count.

4. Experimental Results

To conduct our experiments, we had to model the shift register based solution by transforming the scan cell dependencies into an equivalent single input shared scan-in architecture. As mentioned in the Section 2, if the mode signals are held constant, the Adaptive Scan decompressor breaks down into M static shared scan-in configurations, where M is the number of modes. Each of these M modes can then be independently modeled into a single input shared scan-in architecture to generate test patterns and study the test data volume and test application time. Figure 6 (c) shows one such modeling of a mode shown in figure 6 (a), where four chains of length 5 each are connected to a scan-input. The color of the scan cells depicts the scan cell dependencies, which in this case transform from vertical to diagonal when a single bit coming in is broadcasted to all the scan cells.

The above transformation also holds true if each output of shift register element is connected to multiple chains. A typical Adaptive Scan implementation has a fanout and fanin in the CODEC say f . It can be easily shown that in that case the diagonal scan cell dependency gets multiplied by the fanout factor f . For example, if $f = 3$ and the shift cycle $j = 2$, the number of dependent scan cells are $3 \cdot (2 + 2) = 12$. A similar concept of creating diagonal dependencies was proposed in [12], where the scan chains were structurally rotated using reconfigurable logic in each

mode to form different configurations. It is to be noted that the results presented in this paper based on the modeling of shift register based TAM are pessimistic as this does not take in to account the biggest strength of Adaptive Scan technology: changing the mode signal on per shift basis.

We conducted our experiments on five industrial designs. To compute the fault coverage and test data volume reduction (TDVR) numbers, we inserted the basic scan and all the modeled shared scan-in modes on the same design. For our experiments, all the designs were modeled for an adaptive scan CODEC with four modes i.e., there are 2 control pins required for each CODEC. Since the data connections from the shift register can be reversed, this results into having eight share scan-in modes available at the top level. For all circuits, data for both forward and backward mode has been presented.

4.1 10-bit TAM with 120 Internal Chains

We first present results on TAM design with a 10-bit shift register and for an Adaptive scan CODEC with the following parameters: 10 scan inputs (SI), where data inputs = 8 and control inputs = 2, 120 internal chains and 10 outputs. As shown in Table 1, the compression obtained using Adaptive scan with the above parameters is very close to the ratio of internal chains to the number of inputs i.e., $120 / (8 + 2) = 12$. When a 10-bit TAM is used to feed all the SIs using only forward direction, we observe small pattern inflation and some fault coverage drop. As discussed in the Section 3, the dual mode of the TAM provides additional modes that break scan cell dependencies with a very small area penalty. The results in Table 1 show that using the forward/backward mode is very efficient. We observe that there is no loss in coverage while significant test data volume reduction (TDVR) is obtained by running the TAM in bidirectional mode. For example, for CKT1, the SAS architecture was able to provide 75X compression. The test application time reduction for SAS is exactly equal to the TDVR. It is to be noted that the coverage in compression mode is higher than the scan mode because the reconfiguration logic also is tested in the compression mode.

4.2 32-bit TAM with 240 Internal Chains

In the previous set of experiments, we set 12X compression ratio for Adaptive Scan CODEC and the TAM width was matched with the CODEC input/output of 10. In the second set of experiments, we investigated the scaling of the TAM while reducing the internal compression ratio of the CODEC to $240 / 32 = 7.5$ and increasing the TAM width to 32. Table 2 presents the results for these parameters. Table 2 shows that with such a scaling, the forward direction alone is not efficient at all and results in significant drop in fault coverage. With both the directions, full fault coverage is obtained although at the cost of significant pattern inflation. With the dual TAM configuration, higher compression was obtained for all the circuits. For some circuits, significantly higher

compression gains were obtained. For e.g. 117X and 107X compression was obtained for Ckt1 and Ckt2, respectively.

5. Conclusions

We presented Scalable Adaptive Scan architecture to address the problem of hierarchical implementation of compression. We discussed how an active TAM can be designed using simple shift registers that enables running ATPG at the top while allowing CODEC insertion at the core level. We also presented experimental data showing on how the compression can also be scaled up by using an active TAM. The proposed solution has also raised new questions about what is the optimum size of the shift register, the ratio of shift register size to CODEC input and internal chains that need further investigation.

6. References

- [1] N. A. Touba, "Survey of test vector compression techniques," *IEEE Design & Test of Computers*, vol. 23, pp. 294–303, July-Aug. 2006.
- [2] E. J. Marinissen et al., "A structured and scalable mechanism for test access to embedded reusable cores," *Proc. of Int. Test Conf.*, pp. 284–293, 1998.
- [3] H. Hu and S. Yihe, "A scalable test mechanism and its optimization for test access to embedded cores," *Proc. of int. Conf. on ASIC*, pp. 773–776, 2001.
- [4] E. Larsson and H. Fujiwara, "System-on-chip test scheduling with reconfigurable core wrappers," *IEEE Trans. on VLSI Systems.*, vol. 14, pp. 305–309, 2006.
- [5] Z. S. Ebadi and A. Ivanov, "Time domain multiplexed tam: Implementation and comparison," *Proc. of Design, Auto. & Test in Europe*, pp. 732–737, 2003.
- [6] A. Sehgal, S. K. Goel, E.J. Marinissen and K. Chakrabarty, "Hierarchy-aware and area-efficient test infrastructure design for core-based system chips," *Proc. of Design, Auto. & Test in Europe*, pp. 285–290, 2006.
- [7] A. Dutta and N. A. Touba, "Using limited dependence sequential expansion for decompressing test vectors," *Proc. of Int. Test Conf.*, pp. 1–9, 2006.
- [8] J. Rajski, J. Tyszer, C. Wang and S. M. Reddy, "Convolutional compaction of test responses," *Proc. of Int. Test Conf.*, pp. 745–754, 2003.
- [9] S. Alampally et al., "Evaluation of entropy driven compression bounds on industrial designs," *Proc. Asian Test Symp.*, pp. 13–18, 2008.
- [10] K.-J. Lee, J.J. Chen, and C.H. Huang, "Using a Single Input to Support Multiple Scan Chains," *Proc. Int. Conf. on Computer-Aided Design*, pp. 74–78, 1998.
- [11] A. R. Pandey and J. H. Patel, "An incremental algorithm for test generation in Illinois scan architecture based designs," *Proc. of Design, Auto. & Test in Europe*, pp. 368–375, 2002.
- [12] A. Chandra, H. Yan and R. Kapur, "Multimode Illinois scan architecture for test application time and test data volume reduction," *Proc. VLSI Test Symp.*, pp. 84–92, 2007.
- [13] S. Samaranyake, *A Reconfigurable Shared Scan-In Architecture*, MIT Thesis, May 2003.
- [14] P. Wohl et al., "Minimizing the impact of scan compression," *Proc. VLSI Test Symp.*, pp. 67–74, 2007.
- [15] C. J. Colbourn, A. Rosa, *Triple Systems*, Clarendon Press, Oxford, 1999.
- [16] K.S. Kim and M. Zhang, "Hierarchical test compression for SOC designs," *IEEE Design and Test of Computers*, pp. 142–148, Mar/April 2008.
- [17] V. Iyengar and A. Chandra, "A unified SOC test approach based on test data compression and TAM design," *Proc. of IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems*, pp. 511–518, 2003.
- [18] P. T. Gonciari, P. Rosinger and B. M. Al-Hashimi, "Compression considerations in test access mechanism design," *IEE Proc. of Computers and Digital Techniques*, vol. 152, pp. 89–96, 2005.
- [19] L. Lingappan et al., "Test-volume reduction in systems-on-a-chip using heterogeneous and multilevel compression techniques," *IEEE Trans. on Computer-Aided Design*, vol. 25, pp. 2193–2206, October 2006.

Table 1: TDVR and TATR obtained for 120 internal chains with a 10-bit shift register TAM at the CODEC input.

Circuit	Scan cells	Adaptive Scan Chains =120 SI = 10			Scan SI = 1		SAS For. direction SI = 1		SAS For./Back. direction SI = 1		
		Pat.	FC (%)	TDVR /TATR	Pat.	FC (%)	Pat.	FC (%)	Pat.	FC (%)	TDVR /TATR
Ckt1	54922	1876	95.61	12.6	1688	95.06	2614	95.23	2643	95.58	75.1
Ckt2	118247	2613	98.09	11.8	2563	97.75	4340	97.80	4759	98.05	65.0
Ckt3	58277	3796	94.49	9.2	2905	93.66	11231	93.38	12393	94.67	27.6
Ckt4	26758	3347	91.93	12.0	3357	91.11	5752	89.50	6492	91.62	59.6
Ckt5	41249	1833	97.31	11.3	1595	96.63	4030	96.79	4060	97.33	45.9

Table 2: TDVR and TATR obtained for 240 internal chains with a 32-bit shift register TAM at the CODEC input.

Circuit	Scan cells	Scan SI = 1		SAS For. direction SI = 1		SAS For./Back. direction SI = 1		
		Pat.	FC (%)	Pat.	FC (%)	Pat.	FC (%)	TDVR /TATR
Ckt1	54922	1715	93.62	2862	94.12	3091	95.42	117.7
Ckt2	118247	2562	97.03	4985	97.40	5370	98.08	107.9
Ckt3	58277	2984	92.35	7330	86.37	17142	93.42	37.1
Ckt5	41249	1464	94.15	5527	93.97	6166	97.04	48.5